

Scalar Expressions

A *scalar-expression* consists of a factor or other scalar expressions including scalar operators.

$$\left\{ \begin{array}{c} \left[\begin{array}{c} + \\ - \end{array} \right] \left\{ \begin{array}{c} \text{factor} \\ \text{(scalar-expression)} \end{array} \right\} \\ \text{scalar-expression} \quad \text{scalar-operator} \quad \text{scalar-expression} \end{array} \right\}$$

Concerning reference priority, scalar expressions behave as follows: When a non-qualified variable name is specified in a scalar expression, the first approach is to resolve the variable name as column name of the referenced table. If no column with the specified name is available in the referenced table, Natural tries to resolve this variable as a Natural user-defined variable (host variable).

scalar-operator

$$\left\{ \begin{array}{c} + \\ - \\ * \\ / \\ || \\ \text{CONCAT} \end{array} \right\}$$

A *scalar-operator* can be any of the operators listed above; the operators "-" and "/" must be separated by at least one blank from preceding operators.

factor

$$\left\{ \begin{array}{c} \text{atom} \\ \text{column-reference} \\ \text{aggregate-function} \\ \text{special-register} \\ \text{scalar-function (scalar-expression,...)} \\ \text{scalar-expression unit} \\ \text{case-expression} \end{array} \right\}$$

A *factor* can consist of one of the items listed in the above diagram.

atom

$$\left\{ \begin{array}{l} \textit{parameter} \\ \textit{constant} \end{array} \right\}$$

An *atom* can be either a *parameter* or a *constant*; see also the section Basic Syntactical Items.

column-reference

$$\left[\begin{array}{l} \textit{table-name.} \\ \textit{correlation-name.} \end{array} \right] \textit{column-name}$$

A column-reference is a column name optionally qualified by either a table-name or a correlation-name (see also the section Basic Syntactical Items). Qualified names are often clearer than unqualified names and sometimes they are essential.

Note:

A table name in this context must not be qualified explicitly with an authorization identifier. Use a correlation name instead if you need a qualified table name.

If a column is referenced by a *table-name* or *correlation-name*, it must be contained in the corresponding table. If neither a *table-name* nor a *correlation-name* is specified, the respective column must be in one of the tables specified in the FROM clause.

aggregate-function

$$\left\{ \begin{array}{l} \text{COUNT} \left\{ \begin{array}{l} (*) \\ (\text{DISTINCT } \textit{column-reference}) \end{array} \right\} \\ \left\{ \begin{array}{l} \text{AVG} \\ \text{MAX} \\ \text{MIN} \\ \text{SUM} \end{array} \right\} \left\{ \begin{array}{l} (\text{DISTINCT } \textit{column-reference}) \\ ([\text{ALL}] \textit{scalar-expression}) \end{array} \right\} \end{array} \right\}$$

SQL provides a number of special functions to enhance its basic retrieval power. The so-called SQL aggregate functions currently available and supported by Natural are:

- **AVG** which gives the average of the values in a column,
- **COUNT** which gives the number of values in a column,
- **MAX** which gives the highest value in a column,
- **MIN** which gives the lowest value in a column,
- **SUM** which gives the sum of the values in a column.

Apart from COUNT(*), each of these functions operates on the collection of scalar values in an argument (that is, a single column or a *scalar-expression*) and produces a scalar value as its result.

Example:

```

DEFINE DATA LOCAL
1  AVGAGE   (I2)
END-DEFINE
...
SELECT AVG (AGE)
      INTO AVGAGE
      FROM SQL-PERSONNEL
      ...

```

In general, the argument can optionally be preceded by the keyword DISTINCT to eliminate redundant duplicate values before the function is applied.

If DISTINCT is specified, the argument must be the name of a single column; if DISTINCT is omitted, the argument can consist of a general *scalar-expression*.

DISTINCT is not allowed with the special function COUNT(*), which is provided to count all rows without eliminating any duplicates.

special-register

<div> <div>USER</div> <div> <div>CURRENT TIMEZONE</div> <div>CURRENT DATE</div> <div>CURRENT TIME</div> <div>CURRENT TIMESTAMP</div> <div>CURRENT SQLID</div> <div>CURRENT PACKAGESET</div> <div>CURRENT SERVER</div> </div> </div>

A reference to a *special-register* returns a scalar value.

With the exception of USER, *special-registers* do not conform to standard SQL and are therefore supported by the Natural SQL Extended Set only.

scalar-function

CHAR
COALESCE
DATE
DAY
DAYS
DECIMAL
DIGITS
FLOAT
HEX
HOUR
INTEGER
LENGTH
MICROSECOND
MINUTE
MONTH
NULLIF
SECOND
STRIP
SUBSTR
TIME
TIMESTAMP
TRANSLATE
VALUE
VARGRAPHIC
YEAR

A *scalar-function* is a built-in function that can be used in the construction of scalar computational expressions. The above *scalar-functions* are supported by the Natural SQL Extended Set.

units

YEAR
YEARS
MONTH
MONTHS
DAY
DAYS
HOUR
HOURS
MINUTE
MINUTES
SECOND
SECONDS
MICROSECOND
MICROSECONDS

Units do not conform to standard SQL and are therefore supported by the Natural SQL Extended Set only.

case-expression

$$\text{CASE } \left\{ \begin{array}{l} \text{searched-when-clause} \dots \\ \text{simple-when-clause} \end{array} \right\} \left[\text{ELSE } \left\{ \begin{array}{l} \text{NULL} \\ \text{scalar-expression} \end{array} \right\} \right] \text{END}$$

case-expressions do not conform to standard SQL and are therefore supported by the Natural SQL Extended Set only.

searched-when-clause

$$\text{WHEN } \text{search-condition} \text{ THEN } \left\{ \begin{array}{l} \text{NULL} \\ \text{scalar-expression} \end{array} \right\}$$

See details on *search-condition*.

simple-when-clause

$$\text{scalar-expression } \left\{ \text{WHEN } \text{scalar-expression} \text{ THEN } \left\{ \begin{array}{l} \text{NULL} \\ \text{scalar-expression} \end{array} \right\} \right\} \dots$$